What is claimed is:

1. A method executed in a computer system for performing an operation on a plurality of processors comprising:

issuing a multi-execute command by a host system to a first processor in a first data storage entity connected to the host system, said multi-execute command being a single system call indicating which processors perform at least one operation;

determining whether the first processor is to perform a first operation associated with said multi-execute command;

forwarding said multi-execute command to a second processor in a second data storage entity; and

determining whether the second processor is to perform said first operation associated with said multi-execute command.

2. The method of Claim 1, further comprising:

performing said first operation using at least one of said first processor and said second processor.

3. The method of Claim 1, wherein said first and said second data storage entities are the same.

4. The method of Claim 1, wherein the first and second data storage entities are Symmetrix™ systems.

5. The method of Claim 1, further comprising:

defining a data structure indicating which processors included in said computer system perform said first operation associated with said multi-execute command.

6. The method of Claim 5, further comprising:

determining a path of said multi-execute command, said path including said first and said second processors, said path defining a portion of the processors in said computer system that are forwarded said multi-execute command, said portion including at least two processors; and

wherein said data structure is a bit vector having an entry associated with a corresponding processor included in said portion, each entry in said bit vector having a value of 1 if said corresponding processor is to perform said first operation, and a value of 0 otherwise.

7. The method of Claim 6, wherein said multi-execute command is used to indicate whether each of the processors included in said portion is to perform a plurality of operations, said plurality of operations including said first operation and a second operation different from the first operation; and

wherein said bit vector is a first bit vector and said data structure includes a second bit vector having an entry associated with a corresponding processor included in said portion, each entry in said second bit vector having a value of 1 if said corresponding processor is to perform said second operation, and a value of 0 otherwise.

8. The method of Claim 7, wherein said multi-execute command causes an operation to be performed by a processor in one of a forward path direction and return path direction, said forward path direction being defined as forwarding said multi-execute command from said first processor to an end processor, said return path direction being defined as forwarding said multi-execute command from said end processor to said first processor.

9. The method of Claim 8, further comprising:

performing said first operation on a processor included in said portion in said forward path direction; and

performing said second operation on a processor included in said portion in said return path direction.

10. The method of Claim 9, further comprising:

indicating in said first bit vector whether a processor in said portion is to perform said first operation in said forward path direction; and

indicating in said second bit vector whether a processor in said portion is to perform said second operation in said return path direction.

11. The method of Claim 10, wherein said first operation includes enabling communications of an adapter, and said second operation includes disabling communications of an adapter.

12. The method of Claim 1, wherein shared memory is used for communicating data associated with the multi-execute command between two processors in the same data storage entity, shared memory used to communicate data associated with said multi-execute command.

13. The method of Claim 12, wherein said data includes at least one of: an input parameter, an output parameter, and status information.

14. The method of Claim 13, wherein said status information includes data used in error processing and tracking.

15. The method of Claim 1, further comprising:

performing said first operation between a first pair of two processors of said plurality of processors, at least one processor of said first pair being included in a third data storage entity;

and

performing said first operation between a second pair of two processors of said plurality of processors, at least one processor of said second pair being included in a fourth data storage entity different than said third data storage entity.

16. A computer program product for performing an operation on a plurality of processors in a computer system comprising:

machine executable code for issuing a multi-execute command by a host system to a first processor in a first data storage entity connected to the host system, said multi-execute command being a single system call indicating which processors perform at least one operation;

machine executable code for determining whether the first processor is to perform a first operation associated with said multi-execute command;

machine executable code for forwarding said multi-execute command to a second processor in a second data storage entity; and

machine executable code for determining whether the second processor is to perform said first operation associated with said multi-execute command.

17. The computer program product of Claim 16, further comprising:

machine executable code for performing said first operation using at least one of said first

processor and said second processor.

18. The computer program product of Claim 16, wherein said first and said second data storage entities are the same.

19. The computer program product of Claim 16, wherein the first and second data storage entities are Symmetrix™ systems.

20. The computer program product of Claim 16, further comprising:

a data structure indicating which processors included in said computer system perform said first operation associated with said multi-execute command.

21. The computer program product of Claim 20, further comprising:

machine executable code for determining a path of said multi-execute command, said path including said first and said second processors, said path defining a portion of the processors in said computer system that are forwarded said multi-execute command, said portion including at least two processors; and

wherein said data structure is a bit vector having an entry associated with a corresponding processor included in said portion, each entry in said bit vector having a value of 1 if said corresponding processor is to perform said first operation, and a value of 0 otherwise.

22. The computer program product of Claim 21, wherein said multi-execute command is used to indicate whether each of the processors included in said portion is to perform a plurality of operations, said plurality of operations including said first operation and a second operation different from the first operation; and

wherein said bit vector is a first bit vector and said data structure includes a second bit vector having an entry associated with a corresponding processor included in said portion, each entry in said second bit vector having a value of 1 if said corresponding processor is to perform said second operation, and a value of 0 otherwise.

23. The computer program product of Claim 22, wherein said multi-execute command causes an operation to be performed by a processor in one of a forward path direction and return path direction, said forward path direction being defined as forwarding said multi-execute command from said first processor to an end processor, said return path direction being defined as forwarding said multi-execute command from said end processor to said first processor.

24. The computer program product of Claim 23, further comprising:

machine executable code for performing said first operation on a processor included in said portion in said forward path direction; and

machine executable code for performing said second operation on a processor included in

said portion in said return path direction.

25. The computer program product of Claim 24, further comprising:

machine executable code for indicating in said first bit vector whether a processor in said portion is to perform said first operation in said forward path direction; and

machine executable code for indicating in said second bit vector whether a processor in said portion is to perform said second operation in said return path direction.

26. The computer program product of Claim 25, wherein said first operation includes enabling communications of an adapter, and said second operation includes disabling communications of an adapter.

27. The computer program product of Claim 16, wherein shared memory is used for communicating data associated with the multi-execute command between two processors in the same data storage entity, shared memory used to communicate data associated with said multi-execute command.

28. The computer program product of Claim 27, wherein said data includes at least one of: an input parameter, an output parameter, and status information.

29. The computer program product of Claim 28, wherein said status information includes data used in error processing and tracking.

30. The computer program product of Claim 16, further comprising:

machine executable code for performing said first operation between a first pair of two processors of said plurality of processors, at least one processor of said first pair being included in a third data storage entity; and

machine executable code for performing said first operation between a second pair of two processors of said plurality of processors, at least one processor of said second pair being included in a fourth data storage entity different than said third data storage entity.